



GRADO EN INGENIERÍA INFORMÁTICA

DISEÑO E IMPLEMENTACIÓN DE SISTEMAS DE
INFORMACIÓN

Sistema de soporte a la gestión de la Asignación de Proyectos en Empresas

Autor:
Víctor SÁNCHEZ TERRASA

Tutor académico:
María de los Ángeles LÓPEZ
MALO

Fecha de lectura: 14 de noviembre de 2018
Curso académico 2018/2019

Resumen

En el grado de Ingeniería Informática se necesita un sistema que pueda dar soporte a la gestión de las estancias en prácticas, en estas estancias se incluye también el TFG y la defensa el mismo ante el tribunal, aunque en este caso el alcance del proyecto sólo llegará a la gestionar la asignación de los estudiantes a una estancia en prácticas.

En este documento se encuentra la descripción del proceso de la elaboración del proyecto incluyendo la definición de requisitos, el análisis del diseño de la aplicación, análisis del diseño de la interfaz y la planificación llevada a cabo.

Índice general

1. Introducción	5
1.1. Contexto del proyecto	5
1.2. Motivación del proyecto	5
2. Diseño de la base de datos	7
2.1. Modelo UML proporcionado en clase	7
2.2. Modelo UML definido para nuestro proyecto	9
2.3. Diseño físico	9
3. Diseño de interfaces de usuario.	13
3.1. Planificación	13
3.2. Diseño conceptual	14
3.3. Diseño visual y definición de estilos	14
4. Implementación	27
4.1. Decisiones de implementación	27
4.2. Control de errores	27
4.3. Listado de paquetes y clases	28
5. Conclusiones	31

Capítulo 1

Introducción

1.1. Contexto del proyecto

En el grado de Ingeniería Informática se necesita un sistema que pueda dar soporte a la gestión de las estancias en prácticas, en estas estancias se incluye también el TFG y la defensa el mismo ante el tribunal, aunque en este caso el alcance del proyecto sólo llegará a la gestionar la asignación de los estudiantes a una estancia en prácticas.

1.2. Motivación del proyecto

El objetivo principal, y estratégico, de este producto es proporcionar apoyo a la gestión propia de la asignatura EI1054: Estancias y Trabajo Final de Grado (TFG) de Ingeniería en Informática [1].

A nivel táctico y operativo el producto permite agilizar y mejorar la gestión propia de la asignatura EI1054 (Estancias en prácticas y trabajo final de grado). Proporciona un entorno accesible a las empresas y los supervisores para que la gestión de los TFG se pueda llevar a cabo de manera más colaborativa. Facilita el trabajo de revisión y de aceptación de propuestas de TFG. Mejora la publicación y consulta de las ofertas para la realización de los TFG del grado. Finalmente, proporciona transparencia en el proceso de las asignaciones de estancias y TFG.

SAPE se usa únicamente para gestionar las estancias y el TFG del grado de Ingeniería Informática una vez las empresas han creado una oferta para el grado en el sistema central de gestión de la UJI: el IGLU. SAPE se conecta con IGLU para evitar duplicidades de tareas y facilitar el trabajo de la CEiTFG y los CCG.

En SAPE el usuario principal, y propietarios del producto, es la CEiTFG. Además, son usuarios directos a los alumnos matriculados en la asignatura EI1054, los CCG y las personas de contacto de las empresas. Finalmente, los profesores tutores y los supervisores de las estancias en las empresas son usuarios que sólo reciben información.

SAPE incluye la siguiente funcionalidad:

- Cuando una empresa introduzca una oferta de estancias para EI en IGLU, deberá acceder a SAPE y completar los datos del TFG. La conexión será automática con el mismo usuario y pwd que tienen dado de alta en el IGLU.
- La CEiTFG y los CCG puede aprobar propuestas y solicitar cambios o mejoras. SAPE generará una orden para el envío de mails a los contactos de las empresas. Los mails indica qué información se ha pedido y se registrará los datos del envío para que quede constancia.
- La CEiTFG y los CCG podrán modificar y rechazar propuestas, registrando el motivo. SAPE generará una orden para el envío de un mail que informe la persona de contacto de la empresa del hecho y del motivo, y registrará información del envío para que quede constancia.
- Las personas de contacto de las empresas podrán modificar sus propuestas, mientras no estén aceptadas, usando el mismo usuario y pwd que tienen dado de alta en IGLU.
- En una fecha establecida, la CEiTFG podrá hacer públicas las propuestas revisadas y aprobadas que hage en SAPE hasta esa fecha. Se podrán añadir propuestas aprobadas con posterioridad.
- Los alumnos matriculados en la asignatura podrán consultar las propuestas de proyectos y la información de interés sobre las empresas.
- Los alumnos podrán indicar sus preferencias de manera ordenada incluyendo comentarios.
- La CEiTFG podrá asignar una propuesta de proyecto y un tutor a un alumno. Para poder hacer esta asignación podrán consultar cuál es la ordenación de los alumnos segundo su expediente académico en IGLU.
- Los alumnos podrán consultar la propuesta de asignación provisional y tendrán un plazo para pedir cambios. Estas solicitudes de cambio quedarán registradas junto con la propuesta inicial.
- La CEiTFG podrá revisar los cambios solicitados, hacer cambios y volver a publicar la propuesta.
- La CEiTFG podrá anular una propuesta de asignación para solucionar incidencias puntuales.
- La CEiTFG podrá activar el traspaso al IGLU de las asignaciones aceptadas.

Todas estas funcionalidades no se encuentran en la aplicación, dado que alguna de ellas son complementarias y opcionales dependiendo del número de miembros del grupo escogiendo cuales implementar.

Capítulo 2

Diseño de la base de datos

2.1. Modelo UML proporcionado en clase

Para un correcto desarrollo de la base de datos del proyecto es necesario un modelo UML donde apoyarse. Se parte del modelo del enunciado que se muestra en la Figura 2.1.

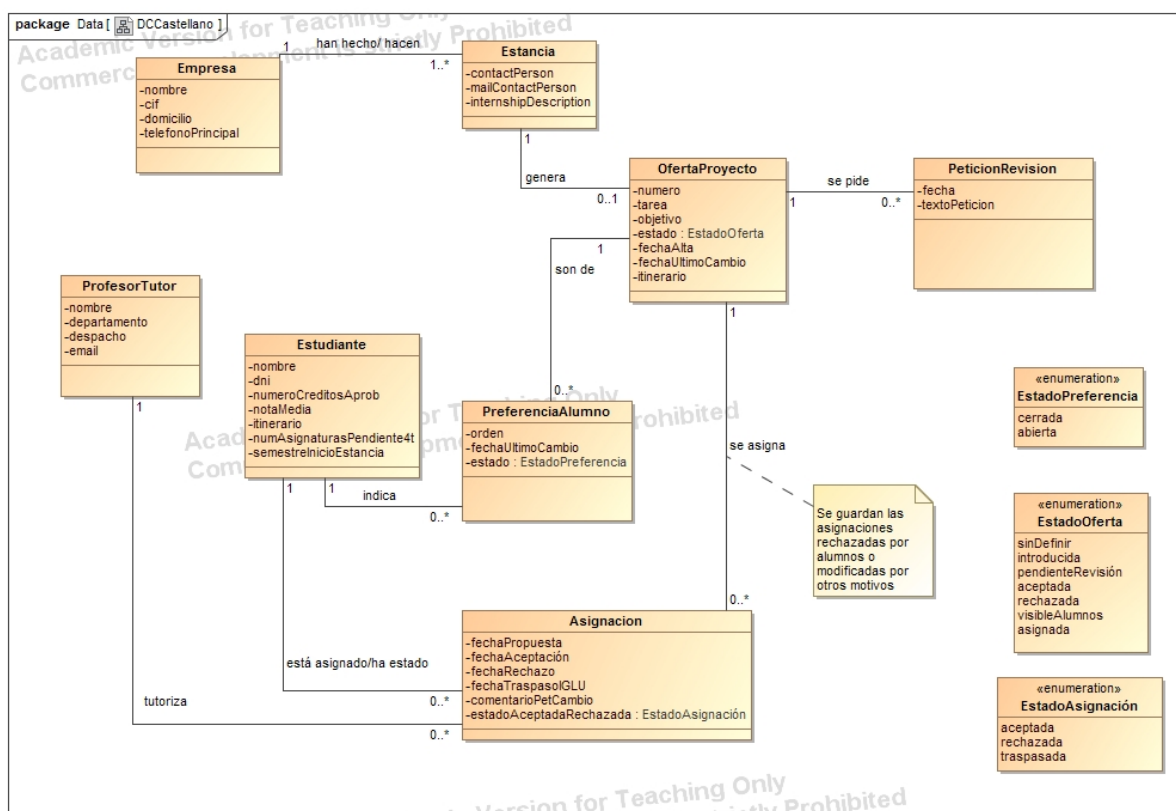


Figura 2.1: UML propuesto en el enunciado [1].

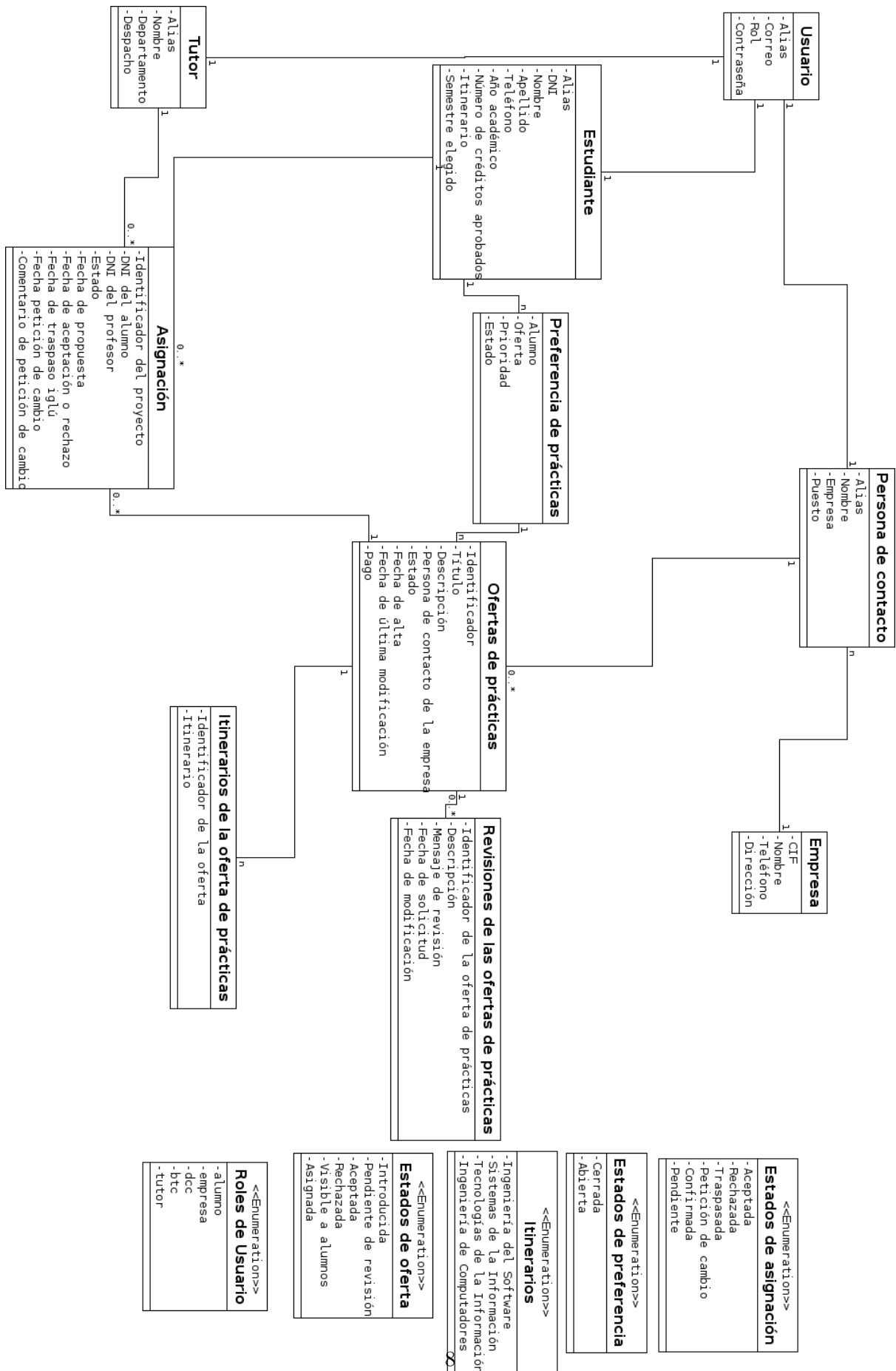


Figura 2.2: UML creado a partir del expuesto en el enunciado.

2.2. Modelo UML definido para nuestro proyecto

Con el modelo ofrecido en el apartado anterior, se ha decidido generar el nuestro propio modelo UML de nuestro proyecto. Como se puede visualizar en la Figura 2.2 nuestro modelo UML tiene cierta semejanza al modelo del enunciado que se muestra en la Figura 2.1.

El mismo modelo UML se también representa el diseño lógico de la base de datos que se emplea en el proyecto donde se encuentran las relaciones entre las tablas, así como los datos primarios y básicos necesarios para el funcionamiento.

2.3. Diseño físico

Partiendo del modelo UML definido, se han escrito los comandos de la base de datos necesarios para el desarrollo del diseño físico que corresponde. En este diseño es necesario que se incluyan todas las clases descritas anteriormente en el diseño lógico además de las relaciones y reglas de borrado y actualización

```
CREATE TABLE Roles_Usuario(  
    nombre VARCHAR(11) NOT NULL,  
    CONSTRAINT pk_rols_usuario PRIMARY KEY (nombre)  
);  
  
CREATE TABLE Usuario(  
    alias VARCHAR(111) NOT NULL UNIQUE,  
    correo VARCHAR(111) NOT NULL,  
    rol VARCHAR(11) NOT NULL,  
    contrasena VARCHAR(555) NOT NULL,  
    CONSTRAINT pk_usuario_alias PRIMARY KEY(alias),  
    CONSTRAINT fk_usuario_rol FOREIGN KEY (rol)  
        REFERENCES roles_usuario(nombre) ON DELETE RESTRICT ON UPDATE CASCADE  
);  
  
CREATE TABLE Empresa(  
    cif VARCHAR(20) NOT NULL UNIQUE,  
    nombre VARCHAR(111) NOT NULL,  
    direccion VARCHAR(222) NOT NULL,  
    telefono VARCHAR(20) NOT NULL,  
    CONSTRAINT pk_empresa_alias PRIMARY KEY (cif)  
);  
  
CREATE TABLE Persona_de_contacto (  
    alias VARCHAR(111) NOT NULL UNIQUE,  
    nombre VARCHAR(333) NOT NULL,  
    empresa VARCHAR(20) NOT NULL,  
    puesto VARCHAR(444) NOT NULL,  
    CONSTRAINT fk_persona_de_contacto_alias FOREIGN KEY (alias)
```

```

REFERENCES usuario(alias) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT fk_persona_de_contacto_empresa FOREIGN KEY (empresa)
REFERENCES empresa(cif) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT pk_persona_de_contacto_alias PRIMARY KEY(alias)
);

CREATE TABLE Tutor (
alias VARCHAR(111) NOT NULL UNIQUE,
nombre VARCHAR(333) NOT NULL,
departamento VARCHAR(222) NOT NULL,
despacho VARCHAR(444) NOT NULL,
CONSTRAINT fk_tutor_alias FOREIGN KEY (alias)
REFERENCES usuario(alias) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT pk_tutor_alias PRIMARY KEY(alias)
);

CREATE TABLE Itinerarios (
tipo VARCHAR(50) NOT NULL,
CONSTRAINT pk_itinerarios PRIMARY KEY (tipo)
);

CREATE TABLE Estudiante (
alias VARCHAR(20) NOT NULL UNIQUE,
dni VARCHAR(12) NOT NULL UNIQUE,
nombre VARCHAR(20) NOT NULL,
apellido VARCHAR(50) NOT NULL,
telefono INTEGER NOT NULL,
anyo_academico SMALLINT NOT NULL,
numero_creditos_aprobados SMALLINT NOT NULL,
itinerario VARCHAR(50) NOT NULL,
semestre_elegido SMALLINT NOT NULL,
CONSTRAINT fk_estudiante_alias FOREIGN KEY (alias)
REFERENCES usuario(alias) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT fk_estudiante_itinerario FOREIGN KEY (itinerario)
REFERENCES Itinerarios(tipo) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT pk_estudiante_alias PRIMARY KEY(alias),
CONSTRAINT ch_estudiante_orden CHECK( 0 < semestre_elegido)
);

CREATE TABLE Estados_de_oferta (
tipo VARCHAR(30) NOT NULL,
CONSTRAINT pk_estados_de_oferta PRIMARY KEY (tipo)
);

CREATE TABLE Ofertas_de_practicas (
id SERIAL NOT NULL,
titulo VARCHAR(111) NOT NULL,
descripcion VARCHAR(99999) NOT NULL,
persona_de_contacto VARCHAR(111) NOT NULL,

```

```

estado VARCHAR(30) NOT NULL,
fecha_alta DATE NOT NULL,
fecha_ultima_modificacion DATE NOT NULL,
pago INTEGER NOT NULL,
CONSTRAINT fk_oferta-de-practicas-persona-de-contacto
    FOREIGN KEY (persona_de_contacto)
    REFERENCES Persona_de_contacto(alias) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT fk_oferta-de-practicas_estado FOREIGN KEY (estado)
    REFERENCES Estados-de-oferta(tipo) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT pk_oferta-de-practicas_id PRIMARY KEY(id),
CONSTRAINT ch_oferta-de-practicas-pago CHECK( 0 <= pago )
);

CREATE TABLE Revisiones_de_la_oferta-de-practicas (
    id_oferta INTEGER NOT NULL,
    descripcion VARCHAR(99999) NOT NULL,
    mensaje_de_revision VARCHAR(99999) NOT NULL,
    fecha_solicitud DATE NOT NULL,
    CONSTRAINT fk_revisiones-de-la-oferta-de-practicas FOREIGN KEY (id_oferta)
        REFERENCES Ofertas-de-practicas(id) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT pk_revisiones-de-la-oferta-de-practicas
        PRIMARY KEY(id_oferta , fecha_solicitud)
);

CREATE TABLE Itinerarios_de_la_oferta-de-practicas (
    id_oferta INTEGER NOT NULL,
    itinerario VARCHAR(99999) NOT NULL,
    CONSTRAINT fk_itinerarios-de-la-oferta-de-practicas
        FOREIGN KEY (id_oferta) REFERENCES Ofertas-de-practicas(id)
        ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT fk_itinerarios-de-la-oferta-de-practicas_itinerarios
        FOREIGN KEY (itinerario) REFERENCES Itinerarios(tipo)
        ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT pk_itinerarios-de-la-oferta-de-practicas
        PRIMARY KEY(id_oferta , itinerario)
);

CREATE TABLE Estados_preferencia-de-practicas (
    tipo VARCHAR(30) NOT NULL,
    CONSTRAINT pk_estados-preferencia-de-practicas PRIMARY KEY (tipo)
);

CREATE TABLE Preferencia (
    alumno VARCHAR(111) NOT NULL,
    id_oferta INT NOT NULL,
    prioridad SMALLINT NOT NULL,
    estado VARCHAR(30) NOT NULL,
    CONSTRAINT fk_preferencia_alias FOREIGN KEY (alumno)
        REFERENCES Estudiante(alias) ON DELETE RESTRICT ON UPDATE CASCADE,

```

```

CONSTRAINT fk_preferencia_ofertas_practicas FOREIGN KEY (id_oferta)
  REFERENCES Ofertas_de_practicas(id) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT fk_preferencia_estado FOREIGN KEY (estado)
  REFERENCES Estados_preferencia_de_practicas(tipo)
  ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT pk_preferencia PRIMARY KEY(alumno, id_oferta),
CONSTRAINT ch_preferencia_prioridad CHECK( 0 < prioridad)
);

CREATE TABLE Estados_de_asignacion (
  tipo VARCHAR(30) NOT NULL,
  CONSTRAINT pk_estados_de_asignacion PRIMARY KEY (tipo)
);

CREATE TABLE Asignacion(
  id_oferta INTEGER NOT NULL,
  alumno VARCHAR(111) NOT NULL,
  tutor VARCHAR(111) NOT NULL,
  estado VARCHAR(40) NOT NULL,
  fecha_de_la_asignacion DATE NOT NULL,
  fecha_de_aceptacion DATE,
  fecha_de_traspaso_iglu DATE,
  fecha_peticion_de_cambio DATE,
  comentario_peticion_de_cambio VARCHAR(99999),
  CONSTRAINT pk_asigna PRIMARY KEY(id_oferta , alumno, tutor),
  CONSTRAINT fk_asigna_id_oferta FOREIGN KEY (id_oferta)
  REFERENCES Ofertas_de_practicas(id) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_asigna_alumno FOREIGN KEY (alumno)
  REFERENCES Estudiante(alias) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_asigna_tutor FOREIGN KEY (tutor)
  REFERENCES Tutor (alias) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_asigna_estado FOREIGN KEY (estado)
  REFERENCES Estados_de_asignacion (tipo)
  ON DELETE RESTRICT ON UPDATE CASCADE
);

```

Capítulo 3

Diseño de interfaces de usuario.

Para el diseño de interfaces de usuario se ha hecho un proceso de diseño centrado en el usuario en el que hemos llevado a cabo los siguientes pasos:

- **Planificación:** En este punto se identifican los objetivos del sitio web, las necesidades, requerimientos y objetivos de los usuarios, aparte de los objetivos técnicos, equipo de trabajo y presupuesto.
- **Diseño conceptual:** Se hizo el sistemas del proyecto desarrollado en el que se ven las diferentes opciones a la hora de navegar por la página web desarrollada según el tipo de rol, que tengas dentro del proyecto ya sea empresa, DCC, BTC o un alumno.
- **Diseño visual y definición de estilos:** se acordó una serie de características visuales que se mantienen durante todas las páginas por las que se pueden navegar a mas de otras características individuales, así como también se ven los distintos test realizados, tantos de usuario como heurísticos.

3.1. Planificación

La planificación se ha inició primero realizando una lectura en grupo para analizar correctamente cuales son los requisitos que requiere el usuario. Para esto se decidió realizar una reunión todos juntos para poder de definir correctamente los objetivos del proyecto y las áreas de actuación de cada uno.

En esta misma reunión se separaron las áreas de actuación para poder repartir equitativamente las horas que requiere el proyecto. La área de actuación de Joshua se trató de la parte de realización del funcionamiento lógico de la aplicación en sincronización con Víctor, mientras Joaquín y Paula se encargan de toda la parte visual. Entre todos se realiza tanto la memoria como la documentación técnica que se ha de entregar posterior mente y Paula se encarga de coordinar el grupo comprobando que las tareas se van realizando semanalmente.

Además de todo esto quedamos en reunirnos todos los miércoles para poner en conjunto

el trabajo realizado. Estas reuniones ha permitido realizar un gráfico en el que se muestra el avance del trabajo lo largo de las semanas. También se crea un grupo de whatsapp para la comunicación y una carpeta de drive para poner en común el trabajo realizado.

Este proyecto al emplearse como recuperación, Víctor se encarga de solucionar los fallos que se encuentran en la actual memoria, de los fallos de diseño y aplicación. Así también se realiza una tutoría con la profesora para comprobar el avance de la memoria y solucionar tanto dudas como fallos cometidos.

3.2. Diseño conceptual

Respecto al diseño conceptual, se pensó en una aplicación que según el rol detectado por el sistema al introducir el usuario, aparezca una página la cual es requerida por el usuario.

A partir de este punto, dependiendo del rol que tiene el usuario, se vuelven a separar las páginas a las que puede acceder el usuario, variando en según el punto en el que se encuentra el usuario.

De una forma más gráfica que encontramos en la Figura 3.1 el mapa web el cual se ha diseñado para toda la plataforma, donde se encuentran las distintas partes a las que se pueden acceder en función del usuario con el que se accede.

3.3. Diseño visual y definición de estilos

A continuación se puede observar tres ejemplos de interfaces que hemos desarrollado para nuestra página web en las Figuras 3.2, 3.3 y 3.4. Estas imágenes representan que es aquello que ve el usuario o el profesor con el rol BTC, viendo así como es el uso de la aplicación y el diseño empleado en la aplicación repetida durante todo el proceso.

El propio equipo y también con la ayuda de familiares y amigos realizó distintos tipos de test enseñados a lo largo del curso, basándose principalmente en el material del profesorado [4].

Primero se expone el análisis heurístico realizado por el propio equipo. Las siguientes Figuras 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16 y 3.17 muestran los resultados obtenidos en el test, mostrando sólo el promedio recibido del total de las pruebas.

Para una evaluación correcta de las interfaces también ha sido necesario la realización de un cuestionario de aceptación de usuarios. En la Figura 3.18 se puede observar el cuestionario realizado junto con el promedio sacado en las respuestas.

A partir de este resultado podemos observar que el resultado no resulta tan negativo, todo lo contrario. Ofrece una buena experiencia al usuario capaz de realizar las tareas de forma deseada.

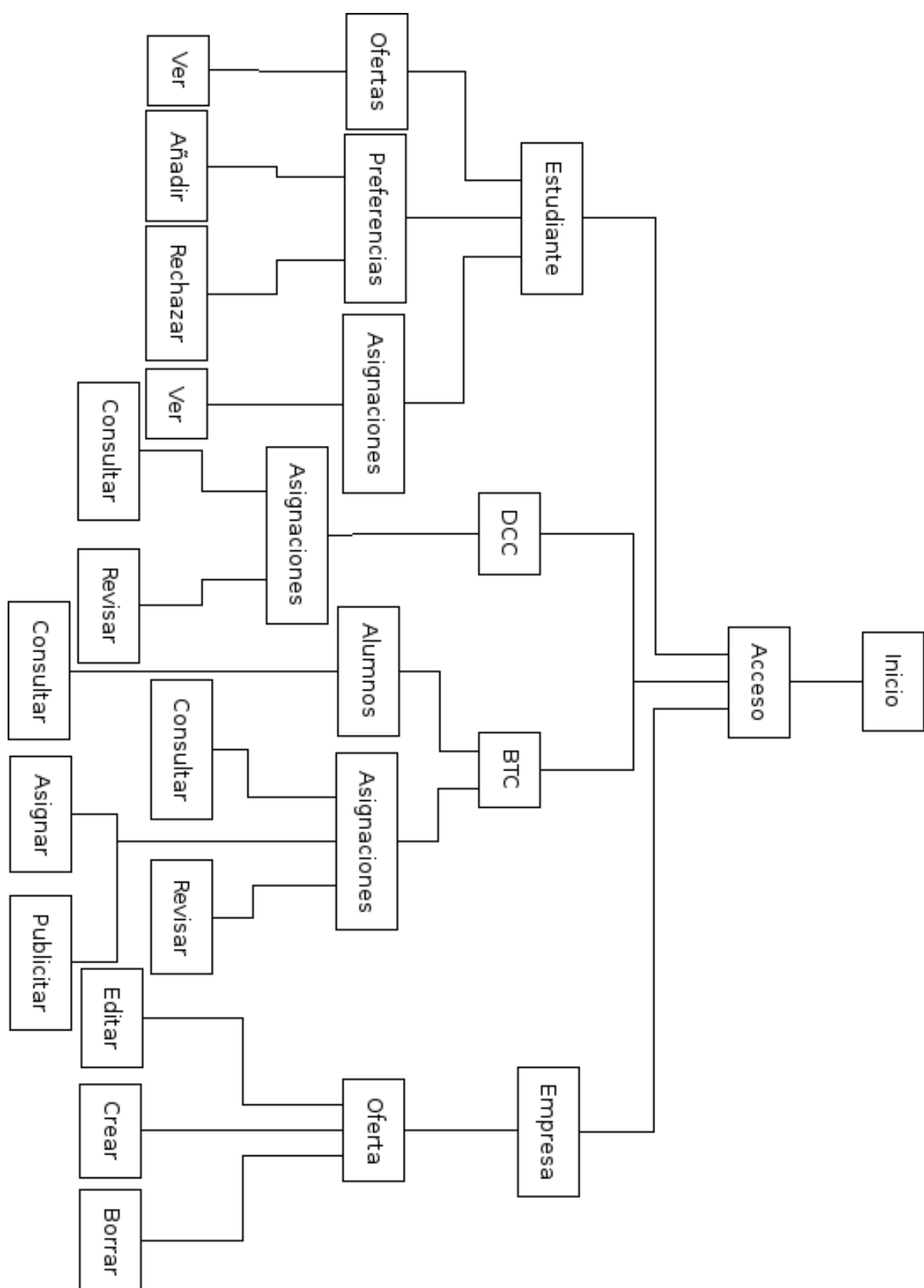


Figura 3.1: Mapa web con el acceso a los distintos contenidos.



Figura 3.2: Esta interfaz permite al alumno puede ver y seleccionar sus preferencias de las prácticas que quiere realizar.



Figura 3.3: Esta interfaz permite al profesor con permisos BTC puede visualizar las prácticas y tener control sobre ellas.

Pràctica assignada

Títol:	IC
Descripció:	IG IC
Nom de l'empresa:	UJI
Encarregat de l'empresa:	Agustina Torrence Nert
Sou:	10€
Estat:	pendiente

Comentari de sol·licitud de canvi:



Aceptar

Solicitar canvi

Figura 3.4: Esta interfaz permite al alumno aceptar o rechazar las prácticas de usuario asignadas por el profesor.

Fecha	03/11/2018
Tipo de conexión	Local
Plataforma	Mozilla Firefoz
Tareas	Aceptación de prácticas, listado de asignaciones y preferencias y listado de prácticas.
Pantallas	Interfaz 1, 2 y 3
Número de análisis	4

Figura 3.5:

Objetivo	Tarea	Pantalla
Conocer si el sistema cumple las expectativas del usuario y se adapta a su contexto social, físico y organizativo.	Aceptación de prácticas, listado de asignaciones y preferencias y listado de prácticas.	Interfaz 1, 2 y 3

Figura 3.6:

Valor	Observaciones
1	Se da la mínima expresión del heurístico en las páginas evaluadas
2	Se da una expresión baja del heurístico en las páginas evaluadas
3	Se da una expresión media del heurístico en las páginas evaluadas
4	Se da una expresión alta del heurístico en las páginas evaluadas
5	Se da la máxima expresión del heurístico en las páginas evaluadas

Figura 3.7:

Generales	Media
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos?	3'75
¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	5
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	4'5
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web?	4'25
¿La estructura general del sitio web está orientada al usuario?	4
¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web?	5
¿Es coherente el diseño general del sitio web?	5
¿Es reconocible el diseño general del sitio web?	5
¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza?	1'75
TOTAL	38'25

Figura 3.8:

Identidad e información	Media
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	5
El logotipo, ¿es significativo, identificable y suficientemente visible?	5
El eslogan o tagline, ¿expresa realmente qué es la empresa y qué servicios ofrece?	5
¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?	1'5
¿Se proporciona mecanismos para ponerse en contacto con la empresa?	2
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	3
En artículos, noticias, informes...¿se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	3'25
TOTAL	24'75

Figura 3.9:

Lenguaje y redacción	Media
¿El sitio web habla el mismo lenguaje que sus usuarios?	5
¿Emplea un lenguaje claro y conciso?	5
¿Es amigable, familiar y cercano?	5
¿1 párrafo = 1 idea?	5
TOTAL	20

Figura 3.10:

Estructura y navegación	Media
La estructura de organización y navegación, ¿es la más adecuada?	4
La estructura de organización y navegación, ¿es la más adecuada?	4
En el caso de ser puramente hipertextual, ¿están todos los nodos comunicados?	4'25
¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos,...)?	5
En menús de navegación, ¿se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística?	5
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace?	5
¿Se ha controlado que no haya enlaces que no llevan a ningún sitio?	5
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación?	5
Las imágenes enlace, ¿se reconocen como clicables? ¿incluyen un atributo 'title' describiendo la página de destino?	3
¿Se ha evitado la redundancia de enlaces?	5
¿Se ha controlado que no haya páginas "huérfanas"?	5
TOTAL	50'25

Figura 3.11:

Layout de la página	Media
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia?	5
¿Se ha evitado la sobrecarga informativa?	5
¿Es una interfaz limpia, sin ruido visual?	5
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	5
¿Se hace un uso correcto del espacio visual de la página?	5
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página?	5
¿Se ha controlado la longitud de página?	3'5
TOTAL	33'5

Figura 3.12:

Elementos multimedia	Media
¿Las fotografías están bien recortadas? ¿son comprensibles? ¿se ha cuidado su resolución?	5
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario?	5
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	5
¿Se ha evitado el uso de animaciones cíclicas?	5
TOTAL	20

Figura 3.13:

Ayuda	Media
Si posee una sección de ayuda, ¿Es verdaderamente necesaria?	1
El enlace a la sección de ayuda, ¿está colocado en una zona visible?	1
¿Se ofrece ayuda contextual en tareas complejas?	1
Si posee FAQs, ¿es correcta tanto la elección como la redacción de las preguntas? ¿y las respuestas?	1
TOTAL	4

Figura 3.14:

Accesibilidad	Media
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	5
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleados facilitan la lectura?	5
¿Existe un alto contraste entre el color de fuente y el fondo?	5
TOTAL	15

Figura 3.15:

Control y retroalimentación	Media
¿Tiene el usuario todo el control sobre el interfaz?	5
¿Se informa constantemente al usuario acerca de lo que está pasando?	5
¿Se informa al usuario de lo que ha pasado?	5
Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema?	5
¿Posee el usuario libertad para actuar?	5
¿Se ha controlado el tiempo de respuesta?	5
TOTAL	30

Figura 3.16:

Heurísticos	Promedio
Generales	4
Identidad e información	3'5
Lenguaje y redacción	5
Estructura y navegación	4.5
Layout de la página	4.5
Elementos multimedia	5
Ayuda	1
Accesibilidad	5
Control y retroalimentación	5

Figura 3.17:

Página web	Promedio		
Lectura en pantalla	difícil	8'6	fácil
Tareas simples	nunca	9'3	siempre
Información organizada	nunca	7'6	siempre
Secuencia de ventanas	correcta	9'1	incorrecta
Terminología e información	Promedio		
Uso de términos adecuados	confuso	9'8	claro
Términos relacionados con las tareas	confuso	10	claro
Posición de los mensajes en la web	correcto	9'4	incorrecto
Indicaciones de entrada de texto	confuso	8'7	claro
Información sobre el progreso	nunca	8'2	siempre
Mensajes de error	confuso	8'4	claro
Aprendizaje	Promedio		
Aprendizaje sobre el uso del sistema	difícil	9'8	fácil
explorar características con prueba y error	confuso	7'8	claro
Recordar nombres y comandos	difícil	8'7	fácil
Tareas sencillas	nunca	8'8	siempre
Mensajes de ayuda	nunca	9'3	siempre
Material de apoyo	nunca	9'2	siempre
Capacidades del sistema	Promedio		
El sistema es rápido	nunca	10	siempre
El sistema es fiable	nunca	10	siempre
El sistema suele ser	difícil	8'2	fácil
Corrección de errores	difícil	7'3	fácil
Diseño para todos los niveles	confuso	9	claro

Figura 3.18:

Capítulo 4

Implementación

4.1. Decisiones de implementación

Durante el diseño de la página web realizada en la asignatura EI1027 - Diseño y implementación de sistemas de información, se ha realizado una estructura MVC (Modelo-Vista-Controlador) en el cual se han desarrollado todas las clases necesarias para el proyecto en las cuales se instancian todos los posibles usuarios de la página web.

También hay que mencionar que se han añadido diferentes roles en la base de datos, ya que dependiendo del rol que se tenga en la página se puede acceder a cierta información.

Por otra parte también se ha realizado varias páginas html, las cuales son las distintas páginas por las que los usuarios podrán navegar dependiendo de qué rol tienen. Por último se ha desarrollado una serie de controladores, los cuales, son los encargados de controlar el acceso a la base de datos.

4.2. Control de errores

Una vez se ha implementado el proyecto ha sido necesario un control de errores que valide los formularios de envío de información, ya que los usuarios no son perfectos y por lo tanto pueden tener errores a la hora de introducir datos en la página.

Para ello se han implementado una serie de validadores para asegurarnos que los datos introducidos por el usuario son válidos y que, en caso de producirse un error durante su introducción, se informe de forma adecuada al usuario para que lo rectifique. Por último también ha sido necesario insertar ciertas restricciones para la correcta actualización de la base de datos.

4.3. Listado de paquetes y clases

En este apartado encontramos las carpetas y ficheros en forma de árbol en la Figura 4.1, empleados en el desarrollo de la aplicación estando separados en las carpetas *modelo*, *dao* y *controller* como bien se ha explicado en el apartado anterior por el *MVC*.

- En el paquete *controller* se encuentran almacenados los controladores de todas las clases y los validadores que se han implementado para la aplicación.
- En el paquete *dao* están las clases que acceden a nuestra base de datos.
- En el paquete *modelo* dispone de las clases que hemos utilizado en nuestro diagrama de clases para la implementación.

```
SAFE
--- java
|   --- es
|       --- uji
|           --- ei1027
|               --- GgSs
|                   --- controller
|                       |   --- AccessController.java
|                       |   --- AsignaController.java
|                       |   --- BtcController.java
|                       |   --- DccController.java
|                       |   --- DccValidator.java
|                       |   --- EmpresaController.java
|                       |   --- EmpresaValidator.java
|                       |   --- EstudianteController.java
|                       |   --- EstudianteValidator.java
|                       |   --- RegistrarEmpresaController.java
|                       |   --- TutorController.java
|                   --- dao
|                       |   --- AsignacionDao.java
|                       |   --- BtcDao.java
|                       |   --- EmpresaDao.java
|                       |   --- EstadoDao.java
|                       |   --- EstudianteDao.java
|                       |   --- ItinerariosDao.java
|                       |   --- Oferta_de_proyectoDao.java
|                       |   --- Oferta_en_listaDao.java
|                       |   --- PersonaDeContactoDao.java
|                       |   --- Preferencia_alumnoDao.java
|                       |   --- RevisionesDao.java
|                       |   --- RolDao.java
|                       |   --- TutorDao.java
|                       |   --- UsuarioDao.java
```

```

|           --- GgSsApplication.java
|           --- GgSsConfiguration.java
|           --- modelo
|               --- Asigna.java
|               --- Btc.java
|               --- Empresa.java
|               --- Estado.java
|               --- Estudiante.java
|               --- Itinerario.java
|               --- ItinerarioOfertaProyecto.java
|               --- Oferta_de_proyecto.java
|               --- PersonaDeContacto.java
|               --- Preferencia_alumno.java
|               --- Revision.java
|               --- Rol.java
|               --- Tutor.java
|               --- UserContacto.java
|               --- Usuario.java
--- resources
    --- application.properties
    --- static
    |   --- css
    |       --- acces.css
    |       --- alumne.css
    |       --- registre_empresa.css
    --- templates
        --- acces.html
        --- alumne
        |   --- confirmacio_semestre.html
        |   --- inicio_asignacion.html
        |   --- listar_ofertas.html
        |   --- seleccio_semestre.html
        |   --- veure_oferta.html
        --- btc
        |   --- alumnos_con_preferencias.html
        |   --- asignaciones.html
        |   --- asignacion_realizada.html
        |   --- asignar_oferta.html
        |   --- confirmar_asignacio_oferta.html
        |   --- index.html
        |   --- ofertas_aceptadas.html
        |   --- ofertas_listadas.html
        |   --- preferencias_alumnos.html
        |   --- ver_asignacio.html
        |   --- veure_oferta.html
        --- dcc
        |   --- confirmar_aceptar.html
        |   --- confirmar_rebutx.html
        |   --- listar_ofertas.html

```

```
|    --- veure_oferta.html
--- empresa
|    --- crear_oferta.html
|    --- editar_oferta.html
|    --- historico.html
|    --- index.html
|    --- veure_oferta.html
--- fragments
|    --- base.html
|    --- header_general.html
--- no_acces.html
--- registre_empresa
    --- contacte_empresa_afegit.html
    --- contacte_empresa.html
    --- index.html
    --- trobar_empresa.html
```

Figura 4.1: Lista de ficheros empleados en el desarrollo de la aplicación empleando el patrón MVC.

Capítulo 5

Conclusiones

Como conclusión hay que decir que el proyecto desarrollado en la asignatura EI1027 - Diseño e Implementación de Sistemas de información ha sido un proyecto que se ha desarrollado durado meses y que ha tenido una carga de trabajo bastante elevada.

Por otro lado es un proyecto muy completo en el que hemos visto todas las etapas para la realización de un sitio web de forma correcta, además de que ha sido un proyecto en el que hemos aprendido muchas herramientas, las cuales, seguro que utilizaremos en un futuro .

Durante el desarrollo de la aplicación, se han adquirido nuevas competencias que se desconocían, como la resolución rápida de conflictos, la capacidad de documentar aplicaciones y las sinergias de trabajar en grupo.

Además la aplicación ha conseguido un funcionamiento bastante mejor del esperado, cumpliendo así los objetivos establecidos e incluidos a lo largo del curso.

Bibliografia

- [1] MARÍA CRISTINA CAMPOS SANCHO, *Cas per al treball de pràctiques de l'assignatura*, https://docs.google.com/document/d/16aIXg8UdvHcTzhVyMVpaHZVP_KefRawoehEIr81dRNc/, 29 de octubre de 2018.
- [2] ISMAEL SANZ, *Introducció a la usabilitat i el disseny d'interfícies*, https://docs.google.com/presentation/d/18kpzN31lZoxK2hXcdZPjD_FLBBdm8KG5-db__kYjlKk/, 29 de octubre de 2018.
- [3] LLEDÓ MUSEROS CABEDO, *Exercici Disseny d'interfícies d'usuari centrat en l'usuari*, <https://docs.google.com/document/d/1g6bx4PCpFXTPRI6hy8UcK9XvFo6obJrpJBEnpC1U1Sk/>, 29 de octubre de 2018.
- [4] LLEDÓ MUSEROS CABEDO E ISMAEL SANZ, *Avaluació de llocs web*, https://docs.google.com/presentation/d/1wMODQ3NZlBf_bLP1PbNOX8dh-wjywwT8cy9EgVYpTh4/, 29 de octubre de 2018.